

Welcome to CFL3D (Version 5.0), a Reynolds-Averaged thin-layer Navier-Stokes flow solver for structured grids. The original version of CFL3D was developed in the early 1980's in the Computational Fluids Laboratory at NASA Langley Research Center; hence the name of the code, which is an acronym for the Computational Fluids Laboratory 3-Dimensional flow solver. As the number of people who have utilized CFL3D has increased over time, so have the demands on the code. Consequently, it is constantly “under construction” with numerous researchers having contributed to code upgrades. Currently, the primary developers/supporters of CFL3D are Dr. Christopher L. Rumsey and Dr. Robert T. Biedron of the Aerodynamics and Acoustics Branch at NASA Langley. An overview of the history of the code can be found in Rumsey, Biedron, and Thomas.<sup>32</sup>

CFL3D solves the time-dependent conservation law form of the Reynolds-averaged Navier-Stokes equations. The spatial discretization involves a semi-discrete finite-volume approach. Upwind-biasing is used for the convective and pressure terms, while central differencing is used for the shear stress and heat transfer terms. Time advancement is implicit with the ability to solve steady or unsteady flows. Multigrid and mesh sequencing are available for convergence acceleration. Numerous turbulence models are provided, including 0-equation, 1-equation, and 2-equation models. Multiple-block topologies are possible with the use of 1-1 blocking, patching, overlapping, and embedding. CFL3D does not contain any grid generation software. Grids must be supplied extraneously.

Version 5.0 of CFL3D has several additional utilities over earlier versions of the code. Most notably, Version 5.0 has the capability to employ sliding patched-zone interfaces, such as might be required to perform rotor-stator computations, for example. However, it is stressed here that CFL3D has been developed primarily as a tool for external aerodynamics analysis. Its use for internal turbomachinery applications has been only as a basic research code thus far; other Navier-Stokes codes specifically designed for turbomachinery applications, such as ADPAC<sup>22</sup>, UncleTurbo<sup>23</sup>, or ROTOR<sup>29</sup> may be better suited to the analysis of such flows.

The purpose of this user's manual is to describe the code and provide instruction for its usage. Chapter 2 explains the set up and running of the code. Various files available with CFL3D are described and directions for their usage are provided. Step by step instructions are listed for running CFL3D, as well as the pre-processing codes needed for grid-overlapping and patching.

Chapter 3 provides a line-by-line description of all the input parameters utilized in CFL3D. This will probably be the chapter referred to the most. It will aid users in understanding the sample problems as well as in setting up their own problems. While some parameters are strictly case dependent, some are fairly general and recommended values are usually indicated. Certain problems may, of course, require altering the recommended values of these parameters as well.

To avoid the use of feet, meters, pounds, grams, etc. in coding the equations, CFL3D solves the Navier-Stokes equations in “nondimensional” form. Each flow-field parameter is nondimensionalized by reference values. For example, all points on a wing may be nondimensionalized by the chord length of the wing. The nondimensionalizations used in CFL3D are described in Chapter 4.

Chapter 5 explains the file formats. The most important of these is the grid file format, since users will need to translate their grids into an appropriate CFL3D format. The restart file format may be needed for user-designed post-processing programs. This chapter also describes the various output files.

Besides providing general information about boundary condition usage in CFL3D, Chapter 6 discusses the various physical boundary conditions available. It also provides descriptions of the multiple block capabilities in CFL3D, mainly, 1-1 blocking, patching, grid overlapping, and grid embedding. Input examples for the various techniques are provided.

The advantages of using multigrid for convergence acceleration are explained in Chapter 7. The theory behind multigrid and its usage are discussed. The use of multigrid with embedded grids is described. Mesh sequencing is also defined and discussed. Input examples are provided there as well.

Chapter 8 discusses time accurate simulation of unsteady flows using sub-iteration schemes. Two types of sub-iterations are currently implemented in CFL3D. The effects of the different types of sub-iterations, as well as the strategy for pursuing time-accurate computations in general, are described.

Chapter 9 contains several test cases for the user to practice with when learning how to run CFL3D. Two-dimensional sample cases include a single block RAE airfoil case, a grid-overlapping NACA 0012 airfoil case, a patched grid NACA 0012 airfoil case, and a grid-overlapping multielement airfoil case. Also described are examples for a fixed flat plate, a vibrating plate, a multistream nozzle, and a rotor-stator. The three-dimensional test cases include an axisymmetric bump, a single block F-5 wing, a single block ONERA M6 wing, and a single block delta wing.

Finally, Chapter 10 is dedicated to troubleshooting. It is basically a compilation of suggestions for common problems encountered by users over the years. It is recommended that this chapter be read at the onset of any difficulties which occur while using CFL3D. Precious time may be saved by learning from the experiences (and, yes, the mistakes!) of

others. It is prudent at this point to suggest to the user that any problems encountered while using CFL3D be reported to the support personnel so that others may benefit from hitherto unknown experiences that may occur.

There are several appendices at the end of this manual which provide more detailed information about the code. Appendix A describes the governing equations. Appendix B explains the time advancement procedure. The spatial discretizations of the inviscid and viscous fluxes are described in Appendix C. The multigrid algorithm is described in Appendix D. The matrix needed to convert the equations from conservative variables to primitive variables is derived in Appendix E. A step-by-step explanation of how to convert the governing equations from Cartesian coordinates to generalized coordinates is provided in Appendix F. Appendix G shows how the forces and moments are calculated. Appendix H describes the turbulence models available in CFL3D. Appendix I illustrates the angle ( $\alpha$  and  $\beta$ ) definitions used in the code. Appendix J contains a list of all the sub-routines in CFL3D and their purposes. Finally, Appendix K explains the differences between Versions 4.1 and 5.0 of CFL3D.

The last two sections of the manual are source listings. Beginning on page 337, there is a partial list of papers that have been published over the years as CFL3D has developed. The last section of the manual is a list of the references used for this document.

It is appropriate at this point to discuss some of the terminology used in the CFL3D *User's Manual*. Terms, such as grid, zone, mesh, and block, are often used interchangeably in CFD discussions and literature. For clarity in this manual, the following distinctions are defined. The *grid* refers to those sets of points that define the flow field and are generated by the user to be read in by CFL3D. A grid may be one entity or it may be composed of several component grids. These component grids may also be called *grid zones*. The grid zones may communicate with one another through 1-1 blocking, patching, overlapping, or embedding. The term *block* is used for bookkeeping purposes. All component grids are blocks and are assigned a block number. The grid or the set of component grids is often referred to as the *global* grid, encompassing the entire flow field. When the component grids are coarsened for such options as multigrid and mesh sequencing, the coarser levels are labelled as blocks and are also assigned block numbers. *Embedded* grids are finer than the component grid in which they reside, but do not encompass the entire zone. Instead, they are placed in regions known to have high gradients to further resolve those areas.

Another terminology definition that should be clarified is the phrase *free stream*. As used throughout this manual, the words *free stream* imply a reference state. CFL3D was developed primarily as an external flow solver, in which case the appropriate reference state *is* free stream. However, for purely internal flows, the concept of free stream has no meaning and the more general concept of *reference state* should be used.

Further distinctions between terms are required in discussing the methods of communication between blocks. When two blocks share a face or a portion of a face and the grid points correspond point to point, the boundary condition communication set up between

the two blocks is called *1-1 blocking*. The blocks in this case are often referred to as being  $C^0$  *continuous*. Grid *patching*, on the other hand, refers to the boundary condition interpolations set up between blocks that share a common face or portion of a face, but which do not match point to point. For example, a fine grid face of one block could pass flow information to a coarse grid face of another block as long as the two blocks shared the face. Grid *overlapping* has neither the restriction of point to point connectivity or a common face between blocks. For example, a cylindrical “inner” grid could communicate with a cubic “outer” grid even though the grid topologies are completely different. The terms *overset grids* and *chimera grids* are synonymous with overlapped grids.

A distinction should also be made for the methods of time advancement. The term *time accurate* means that the flow is actually tracked in time. For example, if the flow is allowed to advance for ten seconds, the flow conditions at a certain point in the flow field might have ten different values, one at each second. For a *steady state* problem, after convergence is reached, the flow conditions will not change with time.

Here is one final tip for reading this manual. In discussions and examples, input variables are typed in **bold** print. Such items as variable names, array names, lines of actual code, input samples, etc. are printed in the `Courier` font. Note that these conventions do not necessarily convey in the World-Wide-Web version of this manual. The CFL3D internet address is: <http://fmad-www.larc.nasa.gov/~rumsey/CFL3D/cfl3d.html>.

Hopefully, this manual will answer most questions encountered when running CFL3D. If problems arise that are not addressed in the sample inputs or in Chapter 10, please contact

Dr. Christopher L. Rumsey  
(757)864-2165  
[c.l.rumsey@larc.nasa.gov](mailto:c.l.rumsey@larc.nasa.gov)

or

Dr. Robert T. Biedron  
(757)864-2156  
[r.t.biedron@larc.nasa.gov](mailto:r.t.biedron@larc.nasa.gov)

NASA Langley Research Center  
Mail Stop 128  
Hampton, VA 23681-0001

Please also let these representatives know if any “bugs” or other useful insights are discovered during the usage of CFL3D.

Every effort has been made to insure that this document represents the correct usage and theoretical underpinnings of CFL3D. The intention has been to make this manual as complete and as self-consistent as possible. However, no guarantee can be given that this manual covers all possible aspects of the code and its usage or that this manual is without error.

